



Joint Subjective and Objective Data Capture and Analytics for Automotive Applications

Downloaded from: <https://research.chalmers.se>, 2026-04-04 15:46 UTC

Citation for the original published paper (version of record):

Johanson, M., Jalminger, J., Frécon, E. et al (2018). Joint Subjective and Objective Data Capture and Analytics for Automotive Applications. IEEE Vehicular Technology Conference.

<http://dx.doi.org/10.1109/VTCFall.2017.8288366>

N.B. When citing this work, cite the original published paper.

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

Joint Subjective and Objective Data Capture and Analytics for Automotive Applications

Mathias Johanson, Jonas Jalminger Emmanuel Frécon Boel Nelson, Tomas Olovsson Mats Gjertz
Alkit Communications AB RISE SICS Chalmers University of Technology Volvo Car Corporation
Mölnadal, Sweden Kista, Sweden Gothenburg, Sweden Gothenburg, Sweden
{mathias, jonas}@alkit.se emmanuel.frecon@ri.se {boeln, tomasol}@chalmers.se mats.gjertz@volvocars.com

Abstract—In this paper we describe a novel technological framework for capture and analysis of both objective measurement data and subjective user experience data for automotive applications. We also investigate how the framework can be extended to address privacy issues by enforcing a rigorous privacy model called differential privacy. The system under development integrates a telematics system with a smartphone app service architecture and a data-driven analytics framework. The hypothesis is that the framework will improve the opportunities of conducting large scale user trials of automotive functions and services, while improving the quality of collected data. To achieve this, a number of challenges are addressed in the paper, including how to design the subjective data capture mechanisms to be both simple to use yet powerful, how to correlate subjective data with objective measurement data, and how to protect the privacy of users.

I. INTRODUCTION

A key to competitiveness in the automotive industry is to be highly responsive to customer needs and expectations as well as market trends. One way to achieve this is to collect and analyze data from connected vehicles to find out how the customers use the product and how the product performs in different situations. The opportunities to employ data capture and analytics for knowledge-driven product development, whereby engineering and design decisions are made based on hard facts rather than best practices and tacit knowledge is gaining strong momentum in the automotive industry [1]. Sophisticated telematics systems and cloud-based analytics frameworks are emerging for these types of applications [2], but what is generally missing is a good way to couple the collected vehicular data and usage data to customer experience data. How the vehicle and user behaves is only one side of the story, the other being how the user experiences the product or would like to experience the product. The objective data being collected through telematics services therefore need to be complemented with subjective data about the customers' experiences of using the product.

The traditional approach to subjective data collection in the automotive industry is through surveys based on questionnaires and interviews with selected customers. However, this type of subjective data collection is time consuming and the collected data sets are typically difficult to correlate with objective measurement data. What the customer says about the way he or she uses a product does not necessarily correspond

to how he or she actually uses the product, nor with how the user would like to use the product or what new features and services are desired. Furthermore, subjective data quality is commonly low since there is a considerable separation in time and space between actually using a product and responding to a traditional questionnaire. The experience the user had while using the product is easily dimmed, forgotten or altered by the passing of time and change of locale. Moreover, when it comes to advanced active safety and autonomous driving services, the volume and complexity of data that need to be collected is high, so a scalable architecture with a high level of automation is needed for capture and analysis of data.

To overcome these problems, we suggest an approach based on a technological framework for coordinated capture and analysis of both objective and subjective data — the latter through the use of a smartphone app which can present tailored questions to selected users to capture specific information about particular events triggered by conditions detected in each user's vehicle during usage. The subjective data submitted through the app is uploaded to a cloud-based analytics framework where objective data, collected from in-vehicle measurement systems are also available for combined analytics. Since the collected data might be privacy sensitive to users, we also investigate how the data can be collected in a privacy-preserving way. This gives the opportunity to carry out large-scale collection of data and automated data-driven analysis, with much higher information quality and in shorter time compared to traditional approaches, reducing the time to market for new product features and customized service offerings. The concept is illustrated in Figure 1.

A. Target Applications

To explore the opportunities of joint subjective and objective data collection, we have developed a proof-of-concept system targeting primarily active safety applications, but with a great potential to be used for many other automotive applications where subjective user data is important, including climate comfort, noise-vibration-harshness (NVH) and ergonomics.

Since active safety and autonomous driving functions increasingly rely on machine learning algorithms that typically require large volumes of training data, systems that can facilitate the collection of relevant training data sets are very important. Finding the relevant training data sets typically requires

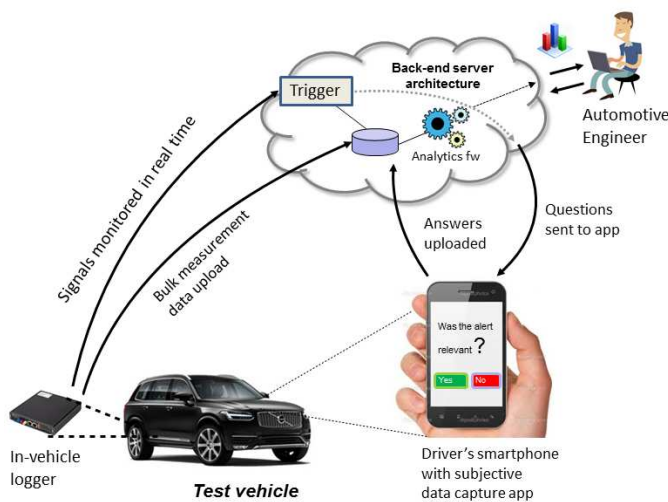


Fig. 1. Joint subjective/objective data capture and analytics concept

human intervention, e.g. to tag or classify whether a given situation belongs to a given category. With the user in the loop through the smartphone app, our approach gives tremendous opportunities to build up a large database of training data. Since the sensor data input to active safety systems typically include complex data types such as video and radar/lidar images, this also affects the design and configuration of the onboard logging devices and corresponding telematics services used to capture and communicate measurement data.

II. CHALLENGES

In the design of the technological framework we have identified a number of challenges that need to be addressed. Some of the more demanding challenges are:

- 1) How can we design the subjective data capture app in a way that makes it easy and safe to use in a vehicle, even while driving?
- 2) How can we design a triggering mechanism to decide when a particular question or set of questions should be posed to a particular user? The triggering mechanism must be versatile and flexible to be usable for all relevant use cases.
- 3) How can we cater for follow-up questions that depend on answers to previous questions?
- 4) How can we protect the privacy of users while at the same time providing automotive engineers with as powerful data collection and analytics tools as possible?

Each of the listed challenges are discussed in the text in the upcoming sections.

III. A FRAMEWORK FOR JOINT SUBJECTIVE-OBJECTIVE DATA CAPTURE AND ANALYTICS

The proof-of-concept framework is composed of the following main components:

- An in-vehicle data capture and telematics system, making it possible to monitor and transmit in-vehicle (CAN bus) signals,

- A cloud-based server infrastructure, including database storage, web-based user interface front-end, and application programming interfaces to provide controlled access to the information resources and framework services,
- A smartphone app to which questions to vehicle users can be pushed from the server infrastructure, and answers recorded and uploaded to the database,
- An analytics service architecture, enabling automated data-driven analysis of data originating from connected vehicles and smartphone apps,
- A app questionnaire authoring tool for designing the questions to be sent to users of the smartphone app,
- A concept for a privacy-preserving framework based on differential privacy.

An overview of the software architecture of the system is shown in Figure 2.

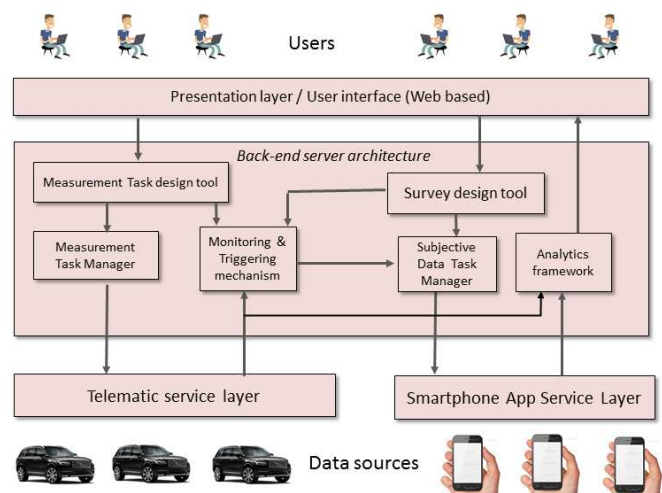


Fig. 2. Software Architecture of the framework for joint subjective/objective data capture and analytics

A. Telematics System

The core component of the telematics system (called WICE) is a Linux-based data capture and communication unit installed in vehicles. The unit executes measurement tasks that support data capture both by passive in-vehicle communication bus monitoring and active diagnostic services. The captured data is uploaded to a cloud-based server infrastructure using 2G, 3G or 4G wireless mobile data communication. The telematics unit also provides additional services such as GPS-based positioning and fleet management.

The communication architecture can handle both bulk upload of data and real-time streaming of data without the need to store it on the solid state disks of the telematics units. For most data capture services, measurement data is stored to disk while a data logging task is running, and then pre-processed and uploaded at the end of the vehicle's driving cycle (i.e. at ignition-off). The streaming mode is used for time-sensitive applications, such as positioning services where it is important to show the current location of moving vehicles.

B. Smartphone App and App Service Architecture

The Smartphone App (see Figure 3) is implemented on top of the Ionic Framework [3] in order to target the most common mobile ecosystems from a single code base. This was deemed necessary in order to rapidly iterate the design throughout the life of the project. Ionic is one of the frameworks making it possible to use regular Web technologies (JavaScript, HTML, CSS, etc.) to develop native-looking apps. A number of specific libraries allow access to local hardware in a manner that hides most of the differences between iOS and Android. There are three major functions provided by the app:

- Registering cars to app user accounts. Car registration is made through manual entry of the car's unique VIN, or through scanning a barcode representing this same identifier and usually printed onto the car's front window.
- Each user account carries a few profile details in order to be able to target specific driver details: persons of above/below average height, in specific age categories, etc.
- Receive and respond to “polls” in order to collect subjective information whenever the back-end has discovered a set of matching metrics that require complementary details for a deeper understanding.

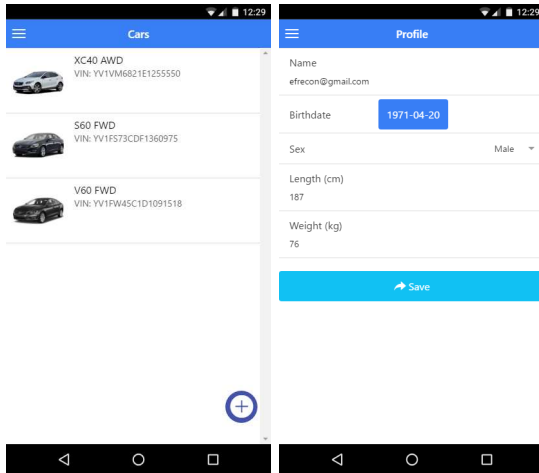


Fig. 3. The screenshot to the left shows the landing page of the smartphone app, listing all cars that are registered to a given account. Note the “hamburger” menu in the top-left corner to access the rest of the app’s functions, and the “+” floating button to trigger car registration. The screenshot to the right shows the profile screen aimed at collecting anthropomorphic data.

Care has been taken to minimise the size and intrusion of the polls as much as possible. Most polls will only contain a few questions, and questions can be conditional, i.e. only asked depending on previous answers within the same poll. The app accepts remote polls even under driving circumstances. However, polls are then read out loud using the mobile platform specific Text-to-Speech (TTS) functions and speech recognition is used to collect answers. Whenever alternatives are offered, small meaningful pieces of these sentences can be used to acknowledge the specific alternative. TTS is also used to inform about errors and progression, so as to engage

the driver in a hands-free dialog. All questions and polls are also present on the smartphone screen, making it possible to answer using touch if necessary or preferred (see Figure 4). The UI uses large, clean and colour-coded buttons to facilitate interaction in all situations, including a bumpy road.

Sometimes it is desirable for polls to be sent in several steps. For example, first as soon as a telematics function has triggered (in order to capture the driver’s answer in the heat of the action), but also later once the car has come to a stop (in order to capture further details about the specific event). These chains are not handled by the polls themselves, but rather through information exchange with the back-end system. Typically, two (or more) polls will be sent by the back-end, possibly conditionally, to capture these situations appropriately. However, the current implementation of the app collects phone position data to approximate speed, and in order to be able to cover these cases without back-end intervention, should that turn out to be necessary in future versions.

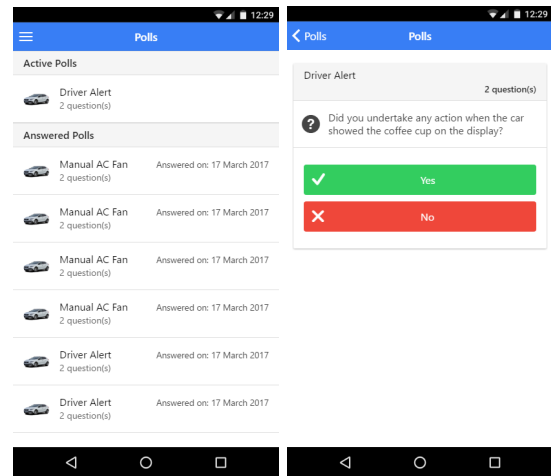


Fig. 4. The screenshot to the left shows the list of polls as seen from the app. Upon answer, polls automatically get sorted into a separate list, shown at the bottom of the screen in order to provide some progress and history feedback. The screenshot to the right shows a typical yes/no question from a poll; the app also supports more elaborate questions with several alternatives. The “coffee cup” is a direct reference to how fatigue alerts are mediated to drivers in the car.

Several drivers/passengers can declare ownership of a single car. At present, relevant polls are sent to all registered users. However, this could be alleviated through automatically detecting which of the registered users is currently (or has just) visited the car. We intend future versions of the app to communicate with the car’s infotainment system as the main source of this type of information. This will also bring the possibility to offer an option that automatically turns on speech recognition (and TTS) when the phone is used in the car. This would prevent polls to be read out loud once the driver has stepped out of the car (which might be disturbing or embarrassing).

C. Back-end Server Architecture and Analytics Framework

The back-end architecture consists of two frameworks. One is the subjective data capture framework described in this

paper which handles the polls and the other is the telematics and analytics framework called WICE [2] which delivers the signals from the car to the back-end data processing framework and provides data processing functions to analyze and visualize the captured data. In order for polls to be delivered to users the person creating the questionnaire must decide upon which set of vehicles should receive the poll when a certain condition occurs and this is done through a web-based tool for creating and managing polls.

The following takes place when a vehicle delivers data to the back-end.

- 1) In-vehicle signals are streamed in real time by the telematics system from connected vehicles to the back-end processing framework. Which signals are streamed is defined in a pre-configured measurement set-up.
- 2) Configurable trigger conditions are evaluated to find whether an event that is of interest has occurred. The trigger conditions are boolean expressions involving signals being streamed, for example `VehicleSpeed > 50 AND Gear=3`. When a trigger condition specified for a specific poll evaluates to true, a service is called which sends the poll to the app which has been registered for the vehicle originating the data stream wherein the interesting event occurred.
- 3) Once the user has answered the poll, the answer is uploaded to the back-end framework and stored in a database, for subsequent analytical processing.

In some cases it is desirable that follow-up questions are posed when the user has responded in a specific fashion. Therefore the back-end framework must be able to evaluate trigger conditions that also include answers to previous polls in order to be able to trigger follow-up polls.

The analytics framework, which is under development, is based on a data-driven approach, whereby data sets uploaded from connected vehicles and apps are automatically analyzed. Analysis results are stored in a knowledge base and made available for visualization, typically as histograms, pie charts or similar.

IV. CASE STUDIES AND USER TRIALS

The technological framework under development will be tested and evaluated in a case study at Volvo Cars wherein two different active safety features are focused: Driver Alert Control (DAC) and Forward Collision Warning (FCW). The DAC system is a driver fatigue detection and warning system. Subjective data is in this case collected to verify whether drivers actually are tired when the DAC system triggers, and to follow up whether they take a break as the system suggests. The FCW system alerts the driver when there is risk for a collision. Subjective data is collected to verify whether issued collision warnings are relevant. The purpose of the case study is to collect subjective user experience data from field trials and to analyze the data together with (objective) measurement data in order to improve the DAC and FCW systems. The hypothesis is that the technological framework presented in this paper will facilitate the orchestration of this kind of

user experience surveys with a potentially large number of participating users, and to improve the quality of the data being collected.

V. PRIVACY ISSUES

While our approach to collect user data opens up new opportunities for improved, data-driven analytics, it also has privacy implications for the drivers that need to be addressed. For example, if a driver has a high number of FCW, it can indicate that the driver is reckless or aggressive, as he or she is often about to collide with objects. An additional privacy issue in this particular setting is that follow-up questions can be issued based on previous answers, which makes the fact that the follow-up question is sent reveal sensitive information. As an example, if a driver ignores the DAC even though he or she is tired, and confesses that this is the case through submitting subjective data during a follow-up poll, this information could be incriminating if the driver is later involved in a traffic accident.

Traditionally, analysts would choose to *de-identify* data, often through removing certain identifiers, such as the vehicle identification number (VIN) and the license plate from the data set. However, real-world examples [4], [5] has shown that de-identification often fails, allowing individuals to be *re-identified*. Examples from the automotive domain where re-identification has been possible include deducing the location of a car based on its speed [6] and fingerprinting drivers from their driving style [7].

In order to protect the driver's privacy, we suggest that data is gathered under *differential privacy*. Differential privacy [8] gives mathematically proven, robust privacy guarantees, which is not provided by any other privacy model. Definition 1 shows the formal definition of differential privacy [9]. Intuitively, differential privacy aims to simulate the best privacy for an individual: namely when he or she has opted out of the analysis. Essentially, differential privacy provides privacy by introducing some inaccuracy, noise, to a real answer. The privacy risk to an individual is then monitored by a privacy budget, which is usually shared by all participants.

Definition 1 (ϵ -differential privacy): A randomized function \mathcal{K} gives ϵ -differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq \text{Range}(\mathcal{K})$,

$$\Pr[\mathcal{K}(D_1) \in S] \leq \exp(\epsilon) \times \Pr[\mathcal{K}(D_2) \in S]$$

To address the privacy issues of the smartphone app, we suggest that a framework¹ for personalized local differential privacy (PLDP) based on randomized response [10] is developed and used when issuing questions from the app. Randomized response is a surveying technique that was invented to avoid evasive answers, for example by lying, from respondents. Randomized response is implemented by letting the respondent flip a coin to determine whether to lie or to answer truthfully, and if the respondent is asked to lie, he or

¹The use of PLDP in this context is ongoing joint work with Hamid Ebadi and Dave Sands at Chalmers University of Technology

she again flips a coin to determine what to answer. As the one collecting the answer does not know whether the respondent tells the truth or provides the random answer determined by the coin, randomized response is said to give *plausible deniability*. When the analyst wants to perform an analysis on the data, he or she uses Bayes' theorem in order to extract the truthful answers. This way data can be collected without it being possible trace a reply back to a specific individual, and also giving the respondents an incentive not to lie unless the coin tells them to.

To address privacy in our architecture, the PLDP framework would be placed in a privacy preservation layer above the smartphone app service layer, and work as an application programming interface (API) used for the questions in the app. Previously, PLDP has only been investigated theoretically [11], and practical implementations do not yet exist. The updated version of the software architecture is shown in Figure 5.

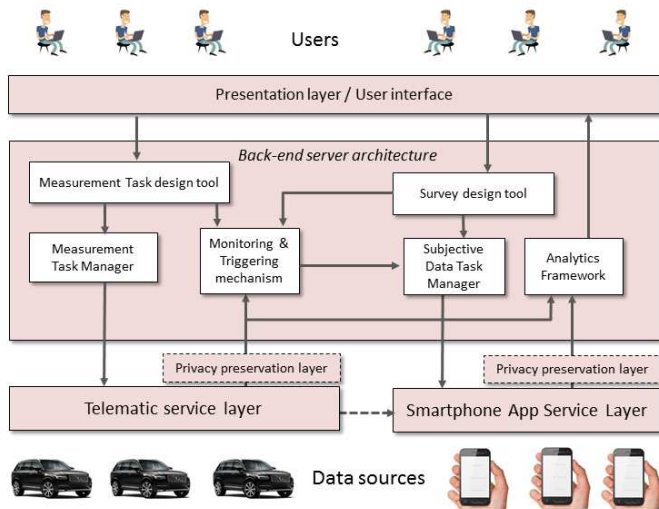


Fig. 5. Updated software architecture of the framework with privacy in mind

Similarly, data from the telematic service layer should also be passed through a privacy preservation layer. The main challenge here is to be able to ask follow-up questions, without letting the back-end server learn the answer to the original questions. Therefore, the polls cannot be issued by the back-end server, but instead will be sent by the telematics server layer as it has access to the car's data. Then, the back-end server chooses a number of cars, uniformly at random, to answer a question. In this way, answers will only be uploaded once the back-end server has chosen that car to participate in a question.

The main implications of a PLDP framework for cars are:

- Local differential privacy does not require a trusted party, as privacy is enforced before the answer is sent to the back-end server. No sensitive data will therefore be uploaded.
- Local differential privacy also gives the driver an incentive not to lie, as raw data never reaches the back-end server.

- Personalized budgets allow for more flexible budgeting than traditional, global budgets, thus allowing for more questions being answered with high accuracy than when using global budgets.
- For each user, a privacy budget needs to be stored and managed, as budgets are personalized.
- Answers to polls need to be saved, in a private state, in the smartphone app.

VI. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper we have explored the opportunities and challenges of joint subjective/objective data capture and analytics for automotive applications. Access to subjective data and sophisticated analytics frameworks in the testing, verification and validation phases of product development promises improved product quality and shorter development cycles, reducing the time to market for new products. We believe that the framework presented in this paper contributes strongly to this. Our future work includes integration of more advanced analytics and visualization mechanisms into the framework and to improve the overall design based on experiences from the case study described in section IV. Furthermore, we have also investigated how to extend the data capture to collect both the subjective user data and the objective car data in a privacy-preserving fashion under differential privacy.

ACKNOWLEDGEMENT

This work was co-funded by VINNOVA, the Swedish Governmental Agency for Innovation Systems.

REFERENCES

- [1] R. Walker, *From Big Data to Big Profits: Success with Data and Analytics*. Oxford University Press, Jul. 2015.
- [2] M. Johanson, S. Belenki, J. Jalminger, M. Fant, and M. Gjertz, "Big Automotive Data: Leveraging large volumes of data for knowledge-driven product development," in *2014 IEEE International Conference on Big Data (Big Data)*, Oct. 2014, pp. 736–741.
- [3] Drifty, "Ionic Framework," visited on 2017-03-28. [Online]. Available: <https://ionicframework.com/>
- [4] P. Samarati and L. Sweeney, "Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression," Technical report, SRI International, Tech. Rep., 1998.
- [5] A. Narayanan and V. Shmatikov, "Robust De-anonymization of Large Sparse Datasets," in *IEEE Symposium on Security and Privacy, 2008. SP 2008*, May 2008, pp. 111–125.
- [6] X. Gao, B. Firner, S. Sugrim, V. Kaiser-Pendergrast, Y. Yang, and J. Lindqvist, "Elastic Pathing: Your Speed is Enough to Track You," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '14. New York, NY, USA: ACM, 2014, pp. 975–986.
- [7] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile Driver Fingerprinting," *Proceedings on Privacy Enhancing Technologies*, vol. 2016, no. 1, pp. 34–50, 2015.
- [8] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *Theory of Cryptography*, ser. Lecture Notes in Computer Science, S. Halevi and T. Rabin, Eds. Springer Berlin Heidelberg, 2006, no. 3876, pp. 265–284.
- [9] C. Dwork, "Differential privacy," in *Automata, languages and programming*. Springer, 2006, pp. 1–12.
- [10] S. L. Warner, "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias," *Journal of the American Statistical Association*, vol. 60, no. 309, pp. 63–69, Mar. 1965.
- [11] R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. Jin, "Private spatial data aggregation in the local setting," in *Data Engineering (ICDE)*. IEEE, 2016, pp. 289–300.